

**IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
TYLER DIVISION**

**JUXTACOMM-TEXAS
SOFTWARE, LLC,**

Plaintiff

vs.

AXWAY, INC., et al.,

Defendants

§
§
§
§
§
§
§
§
§
§

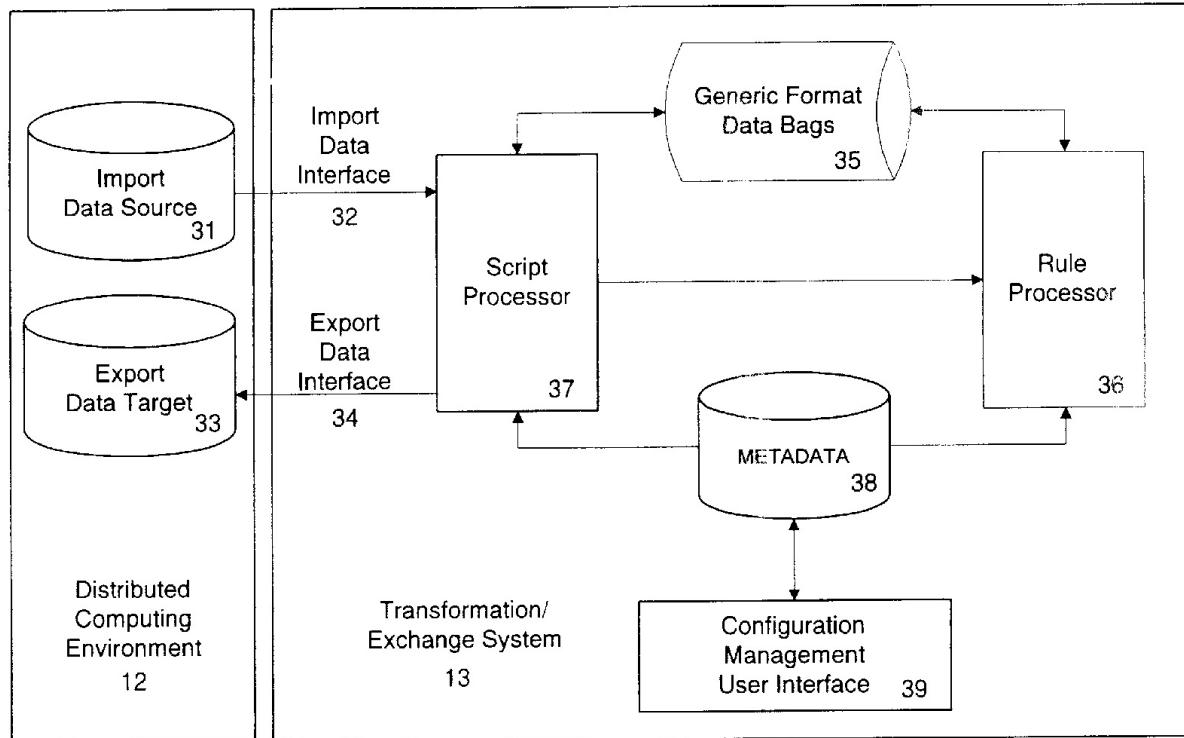
**CASE NO. 6:10CV11
PATENT CASE**

AMENDED MEMORANDUM OPINION

This Memorandum Opinion construes the terms in United States Patent No. 6,195,662 (the '662 patent). The '662 patent was previously construed in *JuxtaComm Technologies, Inc. v. Ascential Software Corporation* ("JuxtaComm I"), Case No. 2:07-CV-00359-LED Docket No. 631.

BACKGROUND

The '662 patent, entitled "System for Transforming and Exchanging Data Between Distributed Heterogeneous Computer Systems," is directed toward the exchange of data between computer systems operating on the basis of different data formats. The '662 patent discloses use of a software-based system that imports data from a source computer system and transforms the data into structures compatible with a target computer system. In other words, the system operates to transform database information from one format into another format, as illustrated by the Figure 2 of the '662 patent:



The patent has two independent claims (Claims 1 and 17) and sixteen dependent claims.¹

The disputed terms are recited in Claims 1 and 17:

1. A distribution system for transforming and exchanging data between heterogeneous computer systems, comprising:
 - a) a systems interface for defining logical import and export data interfaces, data transformation rule sets and scripts;
 - b) a metadata database for storing said logical import and export data interfaces, data transformation rule sets and scripts;
 - c) a script processor for utilizing metadata from the metadata database to control data transformation within said systems interface and movement of said data into and out of said distribution system; and
 - d) a rule set processor responsive to said script processor for manipulating a data bag for storing imported data and a data bag for storing export data.

¹ Claim 13 was cancelled by an *ex parte* reexamination instituted by Microsoft during *JuxtaComm I*. The patentability of Claims 1-12 and 14-19 were confirmed. All disputed claim terms appear in Claims 1 and/or 17.

17. A computer readable memory for transforming and exchanging datastore data between heterogeneous computer systems using different datastore formats for storing similar information, comprising:
 - a) executable code for providing a systems interface for defining logical import and export data interfaces, data transformation rule sets and scripts;
 - b) executable code for providing a script processor for utilizing metadata from a metadata database to control data transformation within said systems interface and movement of said data into and out of said distribution system; and
 - c) executable code for providing a rule set processor responsive to said script processor for manipulating a data bag for storing imported data and a data bag for storing export data.

APPLICABLE LAW

“It is a ‘bedrock principle’ of patent law that ‘the claims of a patent define the invention to which the patentee is entitled the right to exclude.’” *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (en banc) (quoting *Innova/Pure Water Inc. v. Safari Water Filtration Sys., Inc.*, 381 F.3d 1111, 1115 (Fed. Cir. 2004)). In claim construction, courts examine the patent’s intrinsic evidence to define the patented invention’s scope. *See id.*; *C.R. Bard, Inc. v. U.S. Surgical Corp.*, 388 F.3d 858, 861 (Fed. Cir. 2004); *Bell Atl. Network Servs., Inc. v. Covad Commc’ns Group, Inc.*, 262 F.3d 1258, 1267 (Fed. Cir. 2001). This intrinsic evidence includes the claims themselves, the specification, and the prosecution history. *See Phillips*, 415 F.3d at 1314; *C.R. Bard, Inc.*, 388 F.3d at 861. Courts give claim terms their ordinary and accustomed meaning as understood by one of ordinary skill in the art at the time of the invention in the context of the entire patent. *Phillips*, 415 F.3d at 1312–13; *Alloc, Inc. v. Int’l Trade Comm’n*, 342 F.3d 1361, 1368 (Fed. Cir. 2003).

The claims themselves provide substantial guidance in determining the meaning of particular claim terms. *Phillips*, 415 F.3d at 1314. First, a term's context in the asserted claim can be very instructive. *Id.* Other asserted or unasserted claims can also aid in determining the claim's meaning because claim terms are typically used consistently throughout the patent. *Id.* Differences among the claim terms can also assist in understanding a term's meaning. *Id.* For example, when a dependent claim adds a limitation to an independent claim, it is presumed that the independent claim does not include the limitation. *Id.* at 1314–15.

“[C]laims ‘must be read in view of the specification, of which they are a part.’” *Id.* (quoting *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 979 (Fed. Cir. 1995) (en banc)). “[T]he specification ‘is always highly relevant to the claim construction analysis. Usually, it is dispositive; it is the single best guide to the meaning of a disputed term.’” *Id.* (quoting *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996)); *Teleflex, Inc. v. Ficosa N. Am. Corp.*, 299 F.3d 1313, 1325 (Fed. Cir. 2002). This is true because a patentee may define his own terms, give a claim term a different meaning than the term would otherwise possess, or disclaim or disavow the claim scope. *Phillips*, 415 F.3d at 1316. In these situations, the inventor's lexicography governs. *Id.* Also, the specification may resolve ambiguous claim terms “where the ordinary and accustomed meaning of the words used in the claims lack sufficient clarity to permit the scope of the claim to be ascertained from the words alone.” *Teleflex, Inc.*, 299 F.3d at 1325. But, “[a]lthough the specification may aid the court in interpreting the meaning of disputed claim language, particular embodiments and examples appearing in the specification will not generally be read into the claims.” *Comark Commc'ns, Inc. v. Harris Corp.*, 156 F.3d 1182, 1187 (Fed. Cir. 1998) (quoting *Constant v. Advanced Micro-Devices, Inc.*, 848 F.2d 1560,

1571 (Fed. Cir. 1988)); *see also Phillips*, 415 F.3d at 1323. The prosecution history is another tool to supply the proper context for claim construction because a patent applicant may also define a term in prosecuting the patent. *Home Diagnostics, Inc., v. Lifescan, Inc.*, 381 F.3d 1352, 1356 (Fed. Cir. 2004) (“As in the case of the specification, a patent applicant may define a term in prosecuting a patent.”).

Although extrinsic evidence can be useful, it is “less significant than the intrinsic record in determining the legally operative meaning of claim language.” *Phillips*, 415 F.3d at 1317 (quoting *C.R. Bard, Inc.*, 388 F.3d at 862). Technical dictionaries and treatises may help a court understand the underlying technology and the manner in which one skilled in the art might use claim terms, but technical dictionaries and treatises may provide definitions that are too broad or may not be indicative of how the term is used in the patent. *Id.* at 1318. Similarly, expert testimony may aid a court in understanding the underlying technology and determining the particular meaning of a term in the pertinent field, but an expert’s conclusory, unsupported assertions as to a term’s definition is entirely unhelpful to a court. *Id.* Generally, extrinsic evidence is “less reliable than the patent and its prosecution history in determining how to read claim terms.” *Id.*

The patents in suit may contain means-plus-function limitations that require construction. Where a claim limitation is expressed in “means plus function” language and does not recite definite structure in support of its function, the limitation is subject to 35 U.S.C. § 112, ¶ 6. *Braun Med., Inc. v. Abbott Labs.*, 124 F.3d 1419, 1424 (Fed. Cir. 1997). In relevant part, 35 U.S.C. § 112, ¶ 6 mandates that “such a claim limitation ‘be construed to cover the corresponding structure . . . described in the specification and equivalents thereof.’” *Id.* (citing

35 U.S.C. § 112, ¶ 6). Accordingly, when faced with means-plus-function limitations, courts “must turn to the written description of the patent to find the structure that corresponds to the means recited in the [limitations].” *Id.*

Construing a means-plus-function limitation involves multiple inquiries. “The first step in construing [a means-plus-function] limitation is a determination of the function of the means-plus-function limitation.” *Medtronic, Inc. v. Advanced Cardiovascular Sys., Inc.*, 248 F.3d 1303, 1311 (Fed. Cir. 2001). Once a court has determined the limitation’s function, “the next step is to determine the corresponding structure disclosed in the specification and equivalents thereof.” *Id.* A “structure disclosed in the specification is ‘corresponding’ structure only if the specification or prosecution history clearly links or associates that structure to the function recited in the claim.” *Id.* Moreover, the focus of the “corresponding structure” inquiry is not merely whether a structure is capable of performing the recited function, but rather whether the corresponding structure is “clearly linked or associated with the [recited] function.” *Id.*

DISPUTED TERMS

“script processor” and “rule set processor responsive to said script processor”

The construction of “script processor” and “rule set processor responsive to set script processor” turn on the same considerations; therefore, the terms are addressed together. The Court previously construed “script processor” in *JuxtaComm I* as a “software component that processes a script.” *JuxtaComm* proposes the *JuxtaComm I* construction and SAS, DataFlux, and Red Hat agree. Progress, IONA, Rotech, Lawson, Seco Tools, Magic, TIBCO, Ricoh, and Pervasive propose “a software component, distinct from the rule set processor, that processes a script.” Vitria proposes “a software component that runs scripts.”

The Court previously construed “rule set processor responsive to said script processor” in *JuxtaComm I* as “software component that processes rule sets responsive to said script processor.” JuxtaComm proposes the *JuxtaComm I* construction and SAS, DataFlux, and Red Hat agree. Progress, IONA, Rotech, Vitria, Lawson, Seco Tools, Magic, TIBCO, and Ricoh propose “a software component distinct from the script processor that processes rule sets responsive to said script processor.”

JuxtaComm argues that the Court previously rejected the argument that the processors must be separate physical components, thus JuxtaComm contends Defendants’ proposed constructions requiring the software component be distinct from the rule set or script processor should be rejected again. Lawson, Magic, SECO, TIBCO, and Ricoh join Progress, Iona, and Rotech argue that the reexamination file history requires a different construction. They argue that during the reexamination, the Examiner defined the claims as requiring separate and distinct script processor and rule processor elements. They further contend that JuxtaComm agreed to the Examiner’s construction because it did not offer any contrary arguments. Vitria argues the term “script processor,” when read in view of the specification, means “a software component that runs scripts.” Thus, the previous construction incorrectly generalizes specification’s description to mere “processing.” Vitria also contends the term “rule set processor” designates a processor distinct from the script processor, similarly citing to the reexamination for support.

At the *Markman* hearing, JuxtaComm agreed to accept Defendants’ proposed construction for “rule set processor responsive to said script processor,” which is: “a software component distinct from the script processor that processes rule sets, responsive to said script processor,” but JuxtaComm contends this agreement is with the understanding that it is not based

on any disavowal of the claim scope made during a reexam. *See* Tr. *Markman* Hrg., Docket No. 594 at 22:25–24:9. In light of this agreement, it is not necessary to amend the construction of “script processor” as Defendants’ proposed amendment would be redundant.

Accordingly, the Court construes “script processor” as it did in *JuxtaComm I*, “software component that processes a script.” The Court construes “rule set processor responsive to said script processor” as agreed by the parties to mean “a software component distinct from the script processor that processes rule sets responsive to said script processor.”

“systems interface”

The Court previously construed “systems interface” in *JuxtaComm I* as “an interface to the distribution system.” *JuxtaComm* proposes the *JuxtaComm I* construction and Defendants Pervasive, Progress, Iona, Rotech, Lawson, Magic, SECO, TIBCO, Ricoh, SAS, Dataflux, and Red Hat agree. Vitria proposes “a component that allows a user or computer system to interact with the distribution system.”

JuxtaComm contends the doctrine of claim differentiation applies, arguing that dependent claim limitations should not be read into the independent claim. Specifically, *JuxtaComm* contends that Vitria’s proposed construction references “a user” improperly, since it is recited as an additional interface in dependent Claim 2. Vitria argues that *JuxtaComm*’s construction is without support as it implies a generic interface to the distribution system, and contends that the claim requires the “systems interface” to be an operative component of the system. Vitria further argues that the specification describes the “configuration management user interface” as performing the functions of the “system interface.”

The differences between Claims 1 and 2 help define “systems interface.” *Id.* Dependent Claim 2 adds the “user” limitation to independent Claim 1, thus, it is presumed that the independent claim does not include the limitation. *See Phillips*, 415 F.3d at 1314–15. Vitria’s construction is overly limiting as it effectively rewrites the claim to a narrower scope. The previous construction is accurate and will not be disturbed.

Accordingly, the Court construes “system interface” as “an interface to the distribution system.”

“[a script processor for utilizing metadata from the metadata database to control data transformation] within said systems interface”

This phrase appears in claims 1(c) and 17(b) of the ‘662 Patent. Specifically, claim 1(c) states:

A distribution system for transforming and exchanging data between heterogenous computer systems, comprising:

c) a script processor for utilizing metadata from the metadata database to control data transformation **within said systems interface** and movement of said data into and out of said distribution system;

Claim 17(b) states:

A computer readable memory for transforming and exchanging datastore data between heterogenous computer systems using different datastore formats for storing similar information, comprising:

b) executable code for providing a script processor for utilizing metadata from a metadata database to control data transformation **within said systems interface** and movement of said data into and out of said distribution system

Juxtacomm first contends that “systems interface” in element 1(c) and 17(b) is the “systems interface” recited in element 1(a) and 17(a), respectively.” The parties appear to agree

that the term “systems interface,” as used in claim elements 1(a), 1(c), 17(a) and 17(b) is the same.

Defendants contend these claims mean what they say—namely that data transformation must occur “within the systems interface.” JuxtaComm argues Defendants’ proposed construction is nonsensical and is the result of Defendants’ divorced reading of the claims. JuxtaComm asserts that when Claim 17, for example, is read as a whole, it clearly describes “a *script processor* for utilizing metadata from the metadata database *to control data transformation within said systems interface.*” (Emphasis plaintiff’s). Thus, JuxtaComm argues that the claim describes that what takes place within the systems interface is not data transformation, but control of data transformation. *See* ‘662 Patent, Claim 1(a) and Figure 1; *see also* Tr. 29:5–8. JuxtaComm further contends that the claim should be read in view of the specification and disclosed embodiments to mean that control of data transformation “can be accomplished through the definition of a script or scripts using the systems interface.”

Defendants respond that using the ordinary meaning of the phrase does not make it nonsensical. Defendants note that the Court has made it clear that a “systems interface” can be a number of things. It can be a user terminal, a computer, etc., but the claims require that whatever the systems interface is, data transformation has to be within that systems interface.

JuxtaComm’s proposed construction is not consistent with the claim language as written. JuxtaComm seeks to rewrite the claims to conform to the preferred embodiment so that the systems interface—through the scripts run by the script processor—controls data transformation. But that is simply not what the claims state and require. The claims state and require that the script processor control data transformation that is occurring within the systems interface

component. Further, the claims make clear that the systems interface serves to define data transformation rule sets that are stored in the metadata database for use by the script processor.

Perhaps the claims should have said “. . . to control data transformation within said distribution system.” However, what they actually say is “. . . to control data transformation within said systems interface.” The Court may not redraft claims, whether to make them operable or to sustain their validity. *Chef Am., Inc. v. Lamb-Weston, Inc.*, 358 F.3d 1371, 1374 (Fed. Cir. 2004) (courts “construe the claim as written, not as the patentees wish they had written it.”). JuxtaComm cannot import the alleged disclosed embodiment into the claims to rewrite the claims. The claim language is not ambiguous. It is susceptible to only one interpretation, that the data transformation shall occur within said systems interface. Accordingly, the Court construes the term as written, requiring data transformation to occur inside the systems interface; and therefore, no construction is necessary.

“Logical import data interface”

This term was not previously construed in *JuxtaComm I*. Plaintiff proposes “one or more of import connection, import view, and import data bag.” Defendants propose “an interface to import data from an import data source to an import data bag.” JuxtaComm argues Defendants’ proposed construction is inconsistent with the claims and the specification. Specifically, Plaintiff contends Defendants’ proposed construction improperly substitutes “data bag” for “distribution system.” Defendants argue that Plaintiff’s proposed construction improperly limits the term to the preferred embodiment.

JuxtaComm is correct that Defendants’ construction is inconsistent with the specification. However, Defendants are also correct that JuxtaComm’s proposed construction improperly

imports limitations from the specification. Accordingly, the Court construes the term as “an interface to import data from an import data source into the distribution system.”

“Logical export data interface”

This term was not previously construed in *JuxtaComm I*. Defendants propose “an interface to export data from an export data bag to an export data target.” Plaintiff proposes “one or more of export connection, export view, and export data bag.” The parties’ proposed constructions for this term have the same fatal flaws as those discussed in the previous section. Thus, the Court construes this term as “an interface to export data from a distribution system to an export data target.”

“distribution system”

The Court previously adopted the parties’ agreed construction of “distribution system” in *JuxtaComm I*. JuxtaComm proposes the *JuxtaComm I* construction, “a computer system for importing data from a source computer system, transforming the imported data and exporting the transformed data to a target computer system.”² Defendants Pervasive, Progress, Iona, Rotech, Lawson, Magic, SECO, TIBCO, Ricoh, SAS, Dataflux, and Red Hat agree to this construction. Vitria proposes “a computer system for transforming data from an input data source into data usable by an output data target which stores data in a different format.”

JuxtaComm argues that Vitria’s proposed construction improperly eliminates the requirement for import and export of data among computer systems. JuxtaComm argues this exclusion is contrary to the specification and claim requirements. Vitria argues the ’662 patent’s specification repeatedly focuses on transforming data between formats and the previous construction omitted this basic aspect of the invention.

² The parties agreed to this construction in *JuxtaComm I*. Accordingly, the dispute raised here was not previously before the Court. See 2:07cv359, Docket No. 631 at Appendix A.

“Distribution system” is used in the preamble of Claim 1 and the claim body of Claims 1 and 17. Thus, the first issue is whether the preamble is limiting. A preamble is not limiting if it does not recite essential structure or steps, or is not necessary to give life, meaning, and vitality to the claim. *Poly-Am., L.P. v. GSE Lining Tech., Inc.*, 383 F.3d 1303, 1309 (Fed. Cir. 2004). A preamble is not limiting “where a patentee defines a structurally complete invention in the claim body and uses the preamble only to state a purpose or intended use for the invention.” *Rowe v. Dror*, 112 F.3d 473, 478 (Fed. Cir.1997). Here, Claim 1’s preamble describes a “distribution system for transforming and exchanging data between heterogeneous computer systems,” and the body of the Claim 1 requires “import and export data interfaces” and “movement of said data into and out of said distribution system.” Claim 17 also uses the term in the body of the claim to define the scope of the invention. The patent uses the term to provide context for the script processor’s capability of moving data, thus it is essential to understand the claim limitations. Accordingly, “distribution system” is a limitation and will be construed.

Claim 17’s preamble recites the aspect of “different datastore formats.” While Claim 1 does not do so similarly, in view of the specification, the previous construction omits “data in a different format.” However, Vitria’s proposed construction focuses entirely on transformation and fails to give full meaning to the term in regard to data import/export. Accordingly, the previous construction in *JuxtaComm I* is modified to specify that the imported data is transformed *into a different format*.

Accordingly, the Court construes “distribution system” as “a computer system for importing data from a source computer system, transforming the imported data into a different format, and exporting the transformed data to a target computer system.”

“script”

The Court previously construed “script” in *JuxtaComm I* as “a group of commands to control data movement into and out of the system, and to control data transformation within the system.” JuxtaComm proposes the *JuxtaComm I* construction and Defendants Progress, IONA, Rotech, SAS, DataFlux, Red Hat, Lawson, Seco Tools, Magic, TIBCO, and Ricoh agree. Vitria proposes “a group of commands defined to control data movement into and out of the system, and to control data transformation within the system.” Pervasive proposes “a program carried out by another program rather than being directly run on the hardware processor comprising a group of commands to control data movement in and out of the system, and to control data transformation within the system.”

JuxtaComm argues that Vitria’s insertion of the word “defined” is superfluous because the “systems interface” element is for “defining . . . scripts.” JuxtaComm contends Pervasive’s construction improperly imports the Court’s rationale from *JuxtaComm I* into the construction. Vitria argues its construction is consistent with the claim language and specification, which supports that the claimed systems interface is for defining scripts. Pervasive seeks to include the Court’s rationale, which noted that “[i]n general, a script is a program that is executed by another program.” Pervasive’s arguments stem from statements made in the Rebuttal Expert Report of Walter G. Rudd served in *JuxtaComm I*. The Report discusses Pervasive’s prior art product, Data Junction, opining it does not anticipate the claims because the “script” in Data Junction contains commands in a conversion specification rather than a “group of commands” pursuant to the Court’s construction. Pervasive contends the Report’s distinction is unclear and confusing and adding the rationale from the Court’s Opinion in *JuxtaComm I* will clarify the term.

The Court's previous construction is sufficiently clear and does not require further definition. An addition of the word "defined" does not serve to further define "script" because a control command must necessarily be "defined" in order to exist. Likewise, while the Court's rationale serves to further illustrate and support the construction, its incorporation is unnecessary to further define the term. Moreover, at the *Markman* hearing, the Court described its practice regarding the use of rationale from a claim construction opinion for cross examination if necessary, and the parties acknowledged that this satisfied any concerns. *See Markman* Tr. at 55. Accordingly, the Court construes "script" as "a group of commands to control data movement into and out of the system, and to control data transformation within the system."

"Metadata [from the metadata database]"

Juxtacomm proposes "definitions for one or more of connections, views, data bag, rule sets and/or scripts." Vitria agrees with Plaintiff's proposed construction. Lawson, Magic, SECO, TIBCO, Rioch, Progress, Iona, and Rotech take no position. SAS, Dataflux, and Red Hat propose "data about data" or alternatively, no construction is needed.

Juxtacomm argues Defendants' proposed construction is vague and overly broad. Defendants argue Juxtacomm's construction improperly limits the term to the five specific types of metadata cited in the '662 patent. Defendants' proposed construction would not be helpful to a jury. However, Juxtacomm's proposed construction imports limitations from the preferred embodiment and should also be rejected. The Court construes the term to mean "a set of data that describes and gives information about other data."

"Metadata database [for storing]"

This term was previously construed in *Juxtacomm I*. Juxtacomm proposes "a database that stores the logical import and export data interfaces, data transformation rule sets and scripts

used by the system” as construed in *Juxtacomm I*. All Defendants except Vitria also agree with the *Juxtacomm I* construction. Vitria proposes “the database that stores the metadata definitions.”

Juxtacomm contends its proposed construction comes directly from Claim 1(b), which recites “a metadata database for storing said logical import and export data interfaces, data transformation rule sets and scripts.” ‘662 Patent, Col. 9:7–10. Vitria acknowledges that the parties “generally agree about what definitions are stored in [the] metadata database.” Vitria Responsive Br. at 19. The claim language generally defines the scope of the limitation as to what is required to be stored in the metadata database. Vitria cites no substantive reason for its proposed changes, and the Court declines to incorporate the changes. Accordingly, “metadata database [for storing]” is construed as “a database that stores the logical import and export data interfaces, data transformation rule sets and scripts used by the system.”

“utilizing metadata from the metadata database”

Juxtacomm proposes “using metadata from a metadata database” as construed in *Juxtacomm I*. Defendants SAS, Dataflux, and Red Hat agree with the previous construction. Defendants Lawson, Magic, SECO, TIBCO, Ricoh, Progress, Iona, and Rotech (the “Progress Defendants”) propose “using script metadata obtained from the metadatabase.” Vitria proposes “using the stored definitions from the metadata database.”

Juxtacomm argues that Defendants’ proposed constructions do not provide clarity. Regarding the Progress Defendants’ proposed construction, Juxtacomm argues that the Court previously made clear that a script is stored in the metadata database. Thus, Defendants’ proposed construction adds nothing. Juxtacomm further argues that Vitria’s proposed inclusion of “stored definitions” is vague and unnecessary. The Progress Defendants assert that the

construction needs clarification, and this clarification is consistent with the Court's prior analysis. Vitria argues its construction is necessary because JuxtaComm's construction would permit multiple separate databases, but the patentee made a disclaimer during a re-exam, stating that a single database is required.

First, nothing in the Progress Defendants' proposed construction provides any additional guidance that would be beneficial to a jury. Second, Vitria's construction is vague, and its disclaimer argument is without merit. Nowhere does Vitria point to or identify any disclaimer by the patentee that would limit JuxtaComm's proposed construction. Vitria only cites to a characterization of the prior art by the examiner. This is insufficient. Thus, the Court adopts its previous construction and construes the term as "using metadata from a metadata database."

"data bag"

JuxtaComm proposes "a data bag is stored in non-persistent memory, is created by the script processor and exists while the script is running, and contains both generic format data and definitions of that data" as construed in *JuxtaComm I*. Defendants Lawson, Magic, SECO, TIBCO, Ricoh, SAS, Dataflux, and Red Hat agree with this construction. Defendants Progress, Iona, and Rotech (the "Progress Defendants") propose "a data container in non-persistent memory which is created by the script processor and exists while the script is running, and which contains both generic format data and the definitions of that data in a generic format." Vitria proposes "a data bag is stored in non-persistent memory, is created by the script processor and exists while the script is running, and contains the data to be manipulated and the data structure definition which are in general format."

Defendants note that in *JuxtaComm I*, the term was initially disputed, but was ultimately agreed upon by the parties and adopted by the Court. Thus, there was no independent analysis of

the term. The Progress Defendants first argue that the prior agreed construction defines the term with the term itself (i.e. “a data bag is...a data bag”). Thus, Defendants propose “a data container.” Defendants are correct that the term should not be defined with the term itself.

Next Defendants argue that the “definitions of that data” must be in a generic format. Juxtacomm counters that because references to “generic format” data describe the requirements already in the previously adopted construction, a new construction would be redundant. Defendants’ proposed additions to the construction are not necessary. The Court construes “data bag” as “a data container stored in non-persistent memory, is created by the script processor and exists while the script is running, and contains both generic format data and definitions of that data.”

“Data transformation”

Juxtacomm and all Defendants except Progress, Iona, and Rotech agree that the term needs no construction. Juxtacomm alternatively proposes “transforming data using rule sets.” The Progress Defendants propose “change of the appearance or format of data without altering its content.” Juxtacomm argues this proposed construction is not supported by the specification and is instead based on a dictionary definition. Defendants argue that Juxtacomm does not provide a definition, rather it describes how the action is performed.

Juxtacomm and the joining Defendants are correct that this term needs no construction. Defendants’ proposal is overly limiting. Juxtacomm’s alternative proposal improperly includes a statement of how the action is performed. No construction is required as this term is readily understandable by a jury.

“Heterogenous computer systems” / “heterogeneous computer systems using different datastore formats”

Juxtacomm argues no construction is necessary. All Defendants except Progress, Iona, and Rotech agree. Juxtacomm alternatively offers “computer systems using different storage formats.” Progress, Iona, and Rotech propose “two or more computer systems having different architectures” / “two or more computer systems having different architectures using different datastore formats.”

Juxtacomm argues that the reference to “architecture” is beyond the scope of the claim term as used in the context of the patent. Juxtacomm further argues the term is not limiting. Defendants argue that the ordinary meaning of the term is that two architectures are required. The term only appears in the preamble of each claim and does not provide context for any limitation set forth in the body of each claim. This claim needs no construction as it is readily understood by a jury.

“datastore data”

This term appears in the preamble of Claim 17. Juxtacomm argues that construction is not necessary, but offers “any type of data that can be stored in a persistent storage system.” Defendants propose “any type of data stored in a persistent storage system.” Thus, the dispute centers around whether data “can be stored” or “is stored.” In support of Defendants’ proposed construction, Vitria, Progress, Iona and Rotech argue that the term “datastore” is defined in the specification as referring “to the storing of any type of data” *See* Docket No. 493 at 24; Docket No. 494 at 24 (citing ‘662 Patent, 2:26–29). However, that section of the specification defines “datastore” not “datastore data.” The preceeding paragraph discusses datastore data and is consistent with the preamble of Claim 17 in that it describes “a system for transforming and exchanging datastore data between heterogeneous computer systems” *See* ‘662 Patent, Col.

2:13–16. The specification and the preamble of Claim 17 both contemplate that datastore data *can be* stored, not that it must be stored. Defendants' proposed construction improperly introduces an extraneous limitation. Accordingly, the Court construes the term as “data that can be stored in a persistent storage system.”

means-plus-function terms

Defendants contend that various “executable code” terms are means-plus-function terms: (1) “[executable code for providing a rule set processor responsive to said script processor] for manipulating a data bag for storing imported data and a data bag for storing export data;” (2) “executable code for providing a systems interface for defining logical import and export data interfaces, data transformation rule sets and scripts;” and (3) “executable code for providing a script processor for utilizing metadata from a metadata database to control data transformation within said systems interface and movement of said data into and out of said distribution system.” Defendants admit none of these claims use the word “means.” This fact triggers a presumption that § 112, ¶ 6 does not apply. *Lighting World, Inc. v. Birchwood Lighting, Inc.*, 382 F.3d 1354, 1358 (Fed. Cir. 2004). However, Defendants argue that these terms claim a functional result rather than sufficient structures or acts for achieving that result and thus § 112, ¶ 6 should apply. Juxtacomm argues that the terms at issue all include a phrase containing the words “executable code for . . .” and this and other courts have held that the recitation of “code” in a claim conveys sufficient structure. Pl. Br. at 20.

These are not means-plus-function terms. The recited code in these claims describes sufficient structure to avoid the application of § 112, ¶ 6. Executable code exists as a physical structure that is embodied on a physical medium such as a memory storage device. The Court has previously held that “computer code” recites sufficient structure to avoid § 112, ¶ 6. *See Aloft*

Media LLC v. Adobe Systems, Inc., 570 F.Supp. 2d 887 (E.D. Tex. 2008). Defendants present no valid basis for applying § 112, ¶ 6, and the Court declines to do so. Accordingly, these terms are not means-plus-function terms.

CONCLUSION

For the foregoing reasons, the Court interprets the claim language in this case in the manner set forth above. For ease of reference, the Court's claim interpretations are set forth in a table as Appendix A.

So ORDERED and SIGNED this 7th day of December, 2011.

A handwritten signature in black ink, appearing to read 'Leonard Davis', written over a horizontal line.

LEONARD DAVIS
UNITED STATES DISTRICT JUDGE

APPENDIX A

Claim Term	Court's Construction
script processor	software component that processes a script
rule set processor responsive to said script processor	[AGREED] a software component distinct from the script processor that processes rule sets responsive to said script processor
systems interface	an interface to the distribution system
[a script processor for utilizing metadata from the metadata database to control data transformation] within said systems interface	No construction required
Logical import data interface	an interface to import data from an import data source into the distribution system
Logical export data interface	an interface to export data from a distribution system to an export data target
distribution system	a computer system for importing data from a source computer system, transforming the imported data into a different format, and exporting the transformed data to a target computer system
script	a group of commands to control data movement into and out of the system, and to control data transformation within the system
Metadata [from the metadata database]	a set of data that describes and gives information about other data
Metadata database [for storing]	a database that stores the logical import and export data interfaces, data transformation rule sets and scripts used by the system
utilizing metadata from the metadata database	using metadata from a metadata database
data bag	a data container stored in non-persistent memory, is created by the script processor and exists while the script is running, and contains both generic format data and definitions of that data
Data transformation	No construction required

Heterogenous computer systems” / “heterogeneous computer systems using different datastore formats	No construction required
datastore data	data that can be stored in a persistent storage system
“executable code” terms	these are not means-plus-function